Leveraging the Linked Data Principles for Electronic Communications

Antoine Zimmermann*
* *INSA-Lyon, LIRIS, UMR5205, F-69621, France*
`antoine.zimmermann@insa-lyon.fr`

***Abstract*—In this position paper, I advocate the use of Semantic Web and Linked Data technologies to improve online communication. In particular, the Linked Data principles allow users to better integrate multiple forms of online communications, bridging the gap between emails, forums, blogging and microblogging and so on. By lifting message content and metadata to standard formats (using RDF, RDFS, OWL and SPARQL) with a rich, well defined semantics, it is expected that finding information in a mixture of fastly growing user-generated content, conversation and social activities will be easier and more uniform all over the Internet.**

***Keywords*-Semantic Web, Linked Data, email, RDF, SPARQL**

Introduction

Although emails are still among the most used means of electronic communications, people are now often using multiple ways of communicating online, from instant messaging to forums to blogging or micro-blogging, SMSs, using status messages, podcasting, Q&A systems. Since people are switching from one communication channel to the other frequently, they tend to have conversations that start on a medium (say exchanging comments on a blog) and end on a different medium (say emails, SMSs or tweets). The history of the conversation becomes very difficult to track, even for the people involved in the conversation.

I present here how Semantic Technologies and the Linked Data principles can be leveraged to help integrating multiple forms of communications online and offline, as well various data and metadata that link to email content. I start with a short description of Semantic Web technologies. I go on with the Linked Data principles and explain how they can be used on email data. Then, I present how these technologies and existing tools can be used to improve email systems and connect them to online communication and data.

I. Semantic Web Technologies

A. The Resource Description Framework

The Resource Description Framework (RDF[1]) is the basic data model used to describe things on the Web of Data. An RDF document consists of a set of triples ⟨*subject predicate object*⟩ which describe statements about the *subject* being related to the *object* through the relation *predicate*. The *subject* can be a Universal Resource Identifier (URI[2]) which denotes a real world thing (*e.g.*, a person, an idea, a Web document) or a blank node (bnode) which behaves like an existentially quantified variable. The *predicate* is necessarily a URI and denotes a relationship. The *object* can be a URI, a bnode or a *literal*. Literals are pieces of data which denote well defined values such as character strings, integers, dates, as opposed to URIs which denote entities that may not be representable in computers (such as ideas or relationships). An RDF document can be understood as a graph where *subjects* and *objects* are vertices and *predicates* are directed arcs. In this paper, I use abbreviated URIs to shorten names, *e.g.*, `rdf:type` means `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`.

B. SPARQL query language

SPARQL[3] is the query language associated with the RDF data model. With SPARQL, one can query multiple RDF data sources by specifying a graph URI, such that it is easy to integrate information distributed over several sites using SPARQL. In its core, SPARQL allows simple graph matching queries and basic algebraic operations like union, join, intersection, optional matching and filters. SPARQL is currently being extended by the W3C to offer path expressions[4], aggregates[5] and an update language[6].

C. RDFS and OWL

RDF Schema[7] extends RDF with schema modelling capabilities. This allows one to describe the vocabulary used in the data, that is, assign a certain role to specific URIs. In particular, it allows one to define classes and their hierarchical relationships, as well as property hierarchies. It can also specify the domain and range of predicates. These schematic statements are used to infer implicit knowledge from known facts. For example, from the following two triples:

```
foaf:knows rdfs:domain foaf:Person .
ex:Joe foaf:knows ex:Jane .
```

[1] http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[2] http://tools.ietf.org/html/rfc3986

[3] http://www.w3.org/TR/rdf-sparql-query/

[4] http://www.w3.org/TR/2010/WD-sparql11-property-paths-20100126/

[5] http://www.w3.org/TR/2011/WD-sparql11-query-20110512/

[6] http://www.w3.org/TR/2011/WD-sparql11-query-20110512/

[7] http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

I can infer that ⟨`ex:Joe rdf:type foaf:Person`⟩. The Web Ontology Language (OWL[8]) extends RDFS even further by including rich logical constructs such as disjointness, union of classes, cardinality restrictions on properties and much more. This allows for elaboration inferences.

II. The Linked Data Principles

The Linked Data principles are simple publishing principles that take advantage of the architecture of the World Wide Web to make the Web of Data a reality. They are declared as follows:

1) use URIs as names for things;
2) use HTTP URIs so that one can look them up;
3) provide useful description of the things when someone look up the URIs, using standards (RDF, SPARQL);
4) provide links to other things by putting URIs from other domains in the descriptions served.

In this section, I detail how these principles can be used and leveraged to improve email systems and interlink email messages with other forms of online communication.

A. Using URIs to name things

This principle is important for having a consistent identification scheme. With URI, identifiers are shared across data source. Contrary to primary keys in databases which may identify one thing in a database and a completely distinct thing in another database, URIs are meant to identify a unique thing all over the Web. The email header can easily be exploited to generate URIs for emails (using Message-ID), for senders and recipients (using their email addresses), although IDs are not uniformly implemented.

B. Using HTTP URIs

It may seem awkward to provide HTTP URIs when there already exist identifiers in the header of emails. However, having HTTP URI allows one to look up these URI using a Web browser or gather information using a Web crawler. Of course, it is not expected that people will publish their personal conversation on the Web, but there are many cases where having emails published online makes sense. As an example, public mailing lists archives (such as the ones of the W3C) can be found online. Moreover, being on the Web does not necessarily means being public. There can be restricted access which prevent unauthorised users to fetch private conversations. Interestingly, with a HTTP URI, emails can be identified for the purpose of tracking replies, forwarding, etc in the mail client, but can also be consulted online using the same identifier. Moreover, with HTTP, users can access emails the same way they access blog comments, tweets, forum threads and so on. This uniformity is very useful for automatic processing of data, indexing and searching online communication.

C. Providing Useful RDF Information

When looking up the URI of an email, a description of it should be provided. Typically, a human using a Web browser would rather get a textual description of the email (its content, the name of the sender and recipients, etc). However, in order to enhance the processing of the data by software agent, an RDF description can be provided too. In HTTP, it is possible to serve different representation of the same resource using content negotiation. For example, a typical Web browser asks for `Content: text/html` and get a Web page containing the text of the email as well as hyperlinks to, e.g., replies and next message in thread. But a Semantic Web browser may ask for `Content: application/rdf+xml` and get a file in RDF/XML, a serialisation format for RDF. The reason to provide a distinct RDF description for machines rather than a unique HTML page is that the semantics of the information captured by the HTML is usually hidden inside the text or the visual layout. Machines are hardly able to understand natural language and often cannot interpret visual content. For instance, for a Web crawler, a hyperlink indicates that there is an unknown relationship between the pages that are linked, but cannot interpret that the relationship is `replies-to` or `author-of`. This can be made explicit in RDF.

D. Providing Links to Other Data

This is probably the most important principle for the improvement of email management and their integration with other online communication. Whereas emails are naturally interlinked via reply and forward relationships, it is also useful to include links to external data. For instance, linking an email to its author's personal profile online (which can be described in RDF using the FOAF ontology, cf. Section III-A). By doing this, information online and offline can be integrated and queried, using the same standard SPARQL (See Section III-B).

III. How to Leverage these Technologies?

A. Vocabularies and Ontologies

Describing email metadata: as a first step to lift email data onto the Web of Data, the structured information of emails should be mapped to RDF vocabularies. This implies finding RDF equivalents of the metadata in the header. This has already been proposed in the SWAML project[9], which provides a tool to convert a dump of emails into RDF using popular vocabularies (SIOC[10], FOAF[11] and DC[12]). SIOC is mostly used to describe user generated content on online community websites, such as forums, blogs, social networking platforms. Fig. 1 describes the main classes of SIOC and existing properties that can relate their instances.

[8] http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

[9] http://swaml.berlios.de/

[10] Semantically Interlinked Online Communities http://sioc-project.org/

[11] Friend of a Friend http://www.foaf-project.org/

[12] Dublin Core Metatdata Initiative http://dublincore.org/

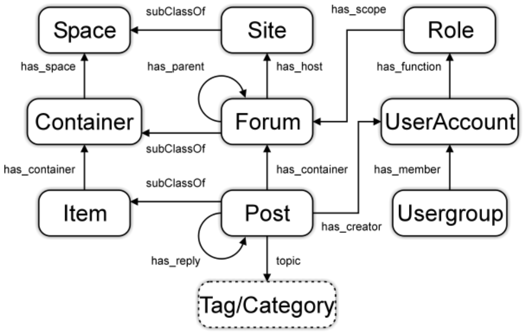

Figure 1. Main SIOC classes

An `Item` corresponds to any online message, be it a blog post, a forum post, a comment, or an email. Items can relate to each other by the property `has_reply`. Authors and recipients are described as user accounts rather than real people, but the persons behind an account can be described using FOAF. DC offers basic metadata such as date, description, format, language, title, etc.

Refining the email description: in addition to the semistructured metadata found in the email header, some information can be extracted automatically or semi-automatically from its content. In [1], we proposed a small extension of SIOC to model quotes in conversations. Quotes are commonly used in emails discussions and forum threads and can generally be identified from the presence of the character '>' at the beginning of a line, and from the HTML structure of forum posts. Having a more fine grained representation of quotes and responses to quotes permits analysing argumentation in a conversation.

Additionally, custom vocabularies could be defined by users with email client extensions in order to tag, interlink or classify emails and contact information. Typically, email clients allow one to define groups of contacts. In a RDF-aware client, these could be modelled as subclasses of `foaf:Person` and organised in a hierarchy. Moreover, for more advanced users, such tools could even allow simple ontological modelling (such as defining properties on certain classes of persons or emails) such that automatic classification would occur from inferences over the custom axioms. However, customised ontologies are subject to change often and non-monotonically because users can make and unmake classes or hierarchical relationships. This has an impact on the kind of reasoners that can be used to operate the classification. In order to avoid costly inferences, it would thus be advisable to limit the expressiveness of custom axioms.[13]

Finally, if emails are related to a specific topic, it would be advisable to reuse existing Web ontologies that are deployed in online datasets. By exploiting the same vocabularies as the ones used online, people can query their emails and online resources together using the same SPARQL queries. Therefore, offering an import mechanism in email clients to enrich the description of the email contents would a plus, although it poses a serious challenge for the development of usable and friendly interfaces.

B. Using SPARQL for Mail Search and Filters

Here I show examples of the use of the graph-based querying facilities that SPARQL offers.

Retrieving emails: sometimes, one may want to retrieve emails in reply to an email which contains a certain sentence or keyword. In current email clients, one can search for keywords, but not for replies to the mails having those keywords. Yet, this can be expressed easily with a SPARQL query:

```
SELECT ?email WHERE {
  ?email sioc:reply_of [ sioc:content ?c ] .
  FILTER regex(?c,"keyword")
}
```

As another example, one may want to retrieve emails that are in a thread, that is, emails that may not be a direct reply to a given email but a reply to a reply, or a further reply. This can be done in SPARQL1.1, which offers property path expressions (see Footnote 4).

```
SELECT ?email WHERE {
  ?email (sioc:reply_of)* [sioc:content ?c] .
  FILTER regex(?c,"keyword")
}
```

Notice the star after `sioc:reply_of`, which means that there can be any number of `reply_of` relationships between `?email` and the node having content `?c`.

Filtering emails: a filter is simply a query evaluated against incoming emails. Therefore, an RDF-aware client, which may become a local SPARQL endpoint, could straightforwardly reuse the queries within the email filter parameters. A filter could also enrich the RDF descriptions by way of CONSTRUCT queries. As an example, let us assume that a user wants to classify emails containing links to IMDb in the custom class `:MovieEmail`. This can be done with the following query, which should be triggered every time an email is received:

```
CONSTRUCT {?email rdf:type :MovieEmail}
WHERE {
    ?email sioc:content ?c ] .
    FILTER regex(?c,"http://www.imdb")
}
```

C. Interlink Emails With Other Online Content

With a model of email representation which encompasses other forms of online communication, it is easy to connect emails to data on the Web such that conversations can seamlessly move from blog posts to comments to tweets, forums then emails. Due to the genericity of the `sioc:reply_to`

[13] The Web Ontology Language (OWL) is very expressive in its unrestricted form and has a high theoretical complexity [2].

predicate, an email can be asserted to be a reply to a blog post or a comment. In pratice, it would be easy to add a link on Web messages to answer them via email rather than through a Web form that would be published online. Similarly, online content may refer to emails. For instance, in W3C working groups, the issue tracker can relate issues and actions to emails sent to the mailing list.[14]

In some email clients, it is possible to assign keywords (or *tags*) to individual emails or threads. Tags are possibly ambiguous words with no formal semantics, but with the Meaning Of A Tag system (MOAT [3]), a tagging system using a MOAT client submits the keywords to a MOAT server which answers with a list of URIs indicating possible meanings for the tag. This way, one can not only disambiguate Apple-the-fruit and Apple-the-brand, but also relate the tagged entity (such as an email) to an entity on the Web of Data, unambiguously identified by a URI.

Finally, a deeper automated analysis of the textual content of emails can be performed to extract meaningful relationships with existing Web entities. For instance, hyperlinks can usefully enrich messages with data retrieved from the Web. In [4], this was done on message board messages, which showed that, even though URLs usually do not link to RDF data, these links can be used to connect the messages to the Web of Data. For instance, a link to IMDb could be used to retrieve RDF data about a movie described in the Linked Movie Database[15]. Using the same approach on mailboxes would certainly lead to similar results.

IV. Discussion and Conclusion

By lifting email data to Semantic Web formats, a mail client becomes a Linked Data platform that can reason over and query email data using established standards. This allows one to integrate local personal conversation with other online sources, or with local data from other applications (calendars, document metadata, etc). Not only this decreases the burden of linking personal data from application to application, but it also allows a seamless integration of Web content inside the personal sphere, along the line of the Nepomuk project[16] with its Social Semantic Desktop [5].[17] However, this should be relativised somehow:

- first, uniformity does not necessarily increase adoption. Recently, Google tried to impose a new single platform called Wave[18] which had the ambition to centralise most of the communication and collaboration tools used by Google aficionados. The project could never reach a critical user mass. Different tools serve different purposes, and removing the distinctions that make the individuality of these tools is, finally, counter-intuitive. However, with Semantic Web standards and Linked Data principles, the distinction is not abolished: an additional compatibility channel is simply added to allow the barriers between applications and datasets to open when and where users need it;
- second, in spite of the promises of semantic technologies, standard formats and ontologies do not erase the interoperability problem: it simply raises it to a higher level, which is addressed by advanced and difficult research fields such as ontology matching, semantic mediation, and so on [7];
- third, I did not address here the foundamental problem of usability from the end user perspective. On the one hand, email systems are not anymore confined to a specialised audience: everybody is likely to use emails in developed countries. On the other hand, Semantic Web technologies are rather complex and understood by proportionally few computer scientists or Web enthusiasts. For these reasons, it is expected that the systems that would implement what is proposed in this paper should hide the technicalities into simplified interfaces. In turn, simplication may lead to a decreased power of the underlying models.

These are important questions that I did not address in this paper, but I hope I offered a good overview of the potential of Linked Data and Semantic Web technologies to the email community.

References

[1] A. Passant, A. Zimmermann, J. Schneider, and J. G. Breslin, "A semantic framework for modelling quotes in email conversations," in *Proc. ISWSA 2010*, 2010.

[2] B. Motik, B. Cuenca-Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz, "OWL 2 Web Ontology Language Profiles," W3C, W3C Recommendation, 2009. [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/

[3] A. Passant, P. Laublet, J. G. Breslin, and S. Decker, "A URI is Worth a Thousand Tags: From Tagging to Linked Data with MOAT," *IJSWIS*, 2009.

[4] S. Kinsella, A. Passant, and J. G. Breslin, "Using hyperlinks to enrich message board content with linked data," in *Proc. I-SEMANTICS 2010*, 2010.

[5] T. Groza, L. Dragan, S. Handschuh, and S. Decker, "Bridging the Gap between Linked Data and the Semantic Desktop," in *Proc. ISWC 2009*, 2009.

[6] S. Scerri, B. Davis, S. Handschuh, and M. Hauswirth, "Semanta - Semantic Email Made Easy," in *Proc. ESWC 2009*, 2009.

[7] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer-Verlag, 2007.

[14] For instance, open issues of the RDF Wroking Group can be found at http://www.w3.org/2011/rdf-wg/track/issues. Issue-2 has two related emails http://www.w3.org/2011/rdf-wg/track/issues/2.

[15] http://www.linkedmdb.org/

[16] http://nepomuk.semanticdesktop.org/

[17] Semanta [6] is a semantic email client in the Nepomuk project which could be extended to leverage the power of Web data.

[18] http://wave.google.com/